

FITUR *DISCOVERY* APLIKASI B.O.S.S (*BUILD OUR SPACE STORE*)

**TUGAS BESAR MATA KULIAH
METODE FORMAL
SE-46-03**



Disusun Oleh:
KELOMPOK 11

- | | |
|---------------------------|--------------|
| 1. Naufal Ajhar El Hafizh | (1302223040) |
| 2. Haikal Alfaro | (1302220098) |
| 3. M. Galang Pradwita | (1302223015) |
| 4. Muhammad Rafi Nadhif | (1302220142) |

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
2023/2024**

BAB 1

PENDAHULUAN

A. Latar Belakang

Fitur discovery atau pencarian adalah fitur yang sangat dibutuhkan dalam sebuah *e-commerce* dan transaksi *online* guna memberikan pelayanan yang maksimal kepada pengguna sehingga pengguna merasa lebih dimudahkan dalam transaksi. Bila pengguna merasa nyaman dan mudah menggunakan aplikasi belanja onlinenya, maka dapat meningkatkan kepercayaan pengguna. Dengan demikian, dapat mendatangkan lebih banyak pengguna dan pada akhirnya akan meningkatkan pertumbuhan bisnis.

B. Tujuan

Fitur discovery adalah fitur yang memungkinkan pengguna aplikasi B.O.S.S untuk menemukan produk-produk yang sesuai dengan kebutuhan dan minat mereka. Fitur ini dapat memberikan rekomendasi produk berdasarkan kriteria tertentu, seperti kategori, harga, lokasi, ulasan, dan lainnya. Fitur ini juga dapat menampilkan produk-produk terbaru, terlaris, atau terpopuler di aplikasi B.O.S.S.

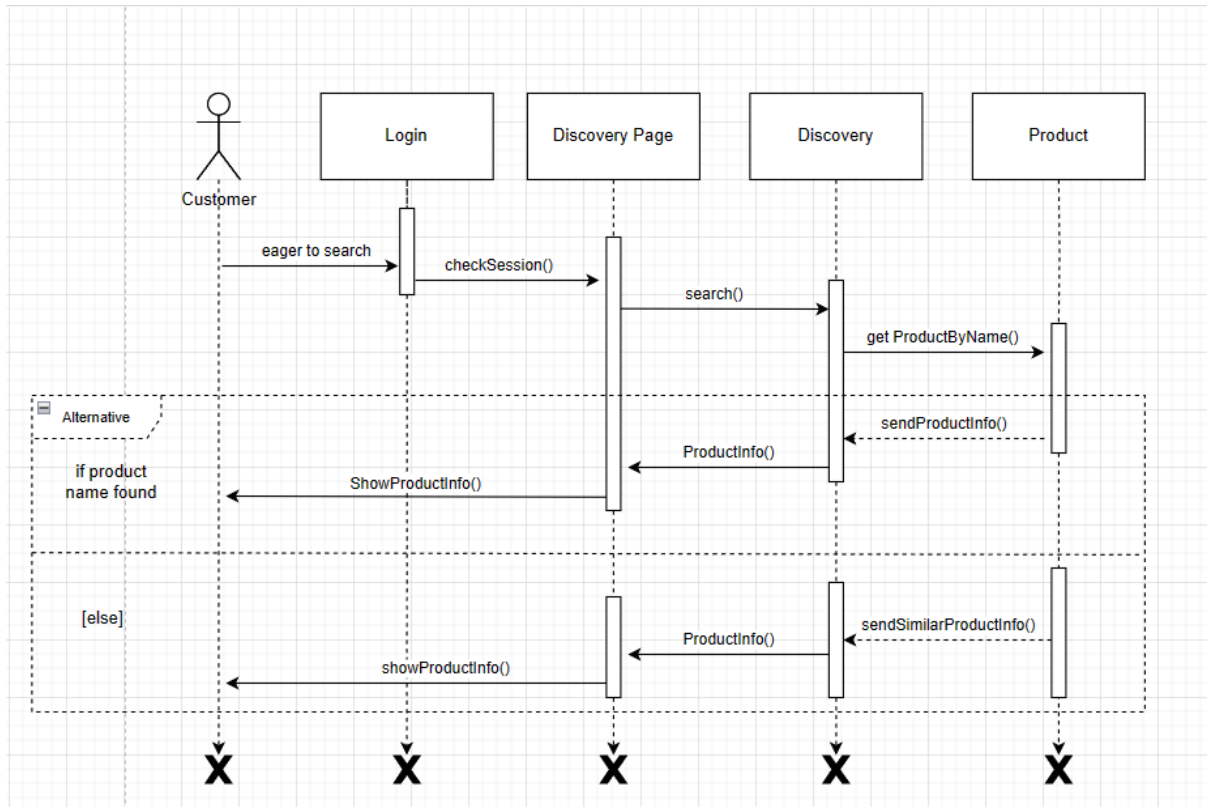
Tujuan dari fitur discovery adalah :

1. Meningkatkan pengalaman pengguna aplikasi toko bangunan, dengan memberikan kemudahan dan kecepatan dalam mencari dan memilih produk yang diinginkan.
2. Meningkatkan penjualan dan loyalitas pelanggan, dengan menawarkan produk-produk yang relevan, berkualitas, dan kompetitif.
3. Membantu pengguna aplikasi toko bangunan untuk menemukan solusi terbaik untuk kebutuhan bangunan mereka.
4. Meningkatkan kepuasan pengguna dengan memberikan rekomendasi produk yang sesuai dengan kebutuhan dan minat mereka.
5. Meningkatkan retensi pelanggan dengan menawarkan produk-produk yang terbaru, terlaris, atau terpopuler di aplikasi toko bangunan.

C. Pemodelan

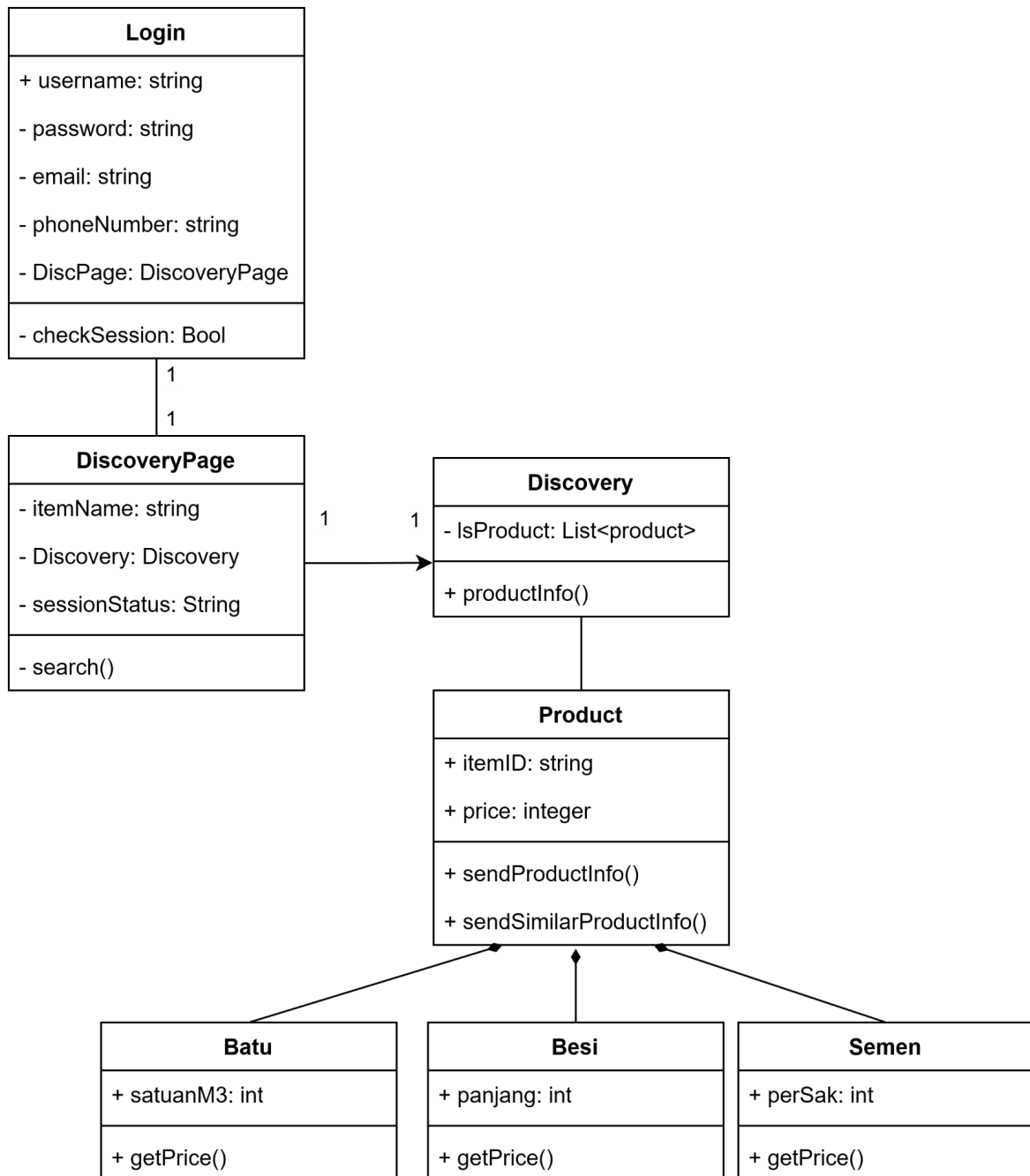
Kami menggunakan pemodelan UML berupa sequence diagram dan class diagram untuk sistem discovery aplikasi e-commerce :

Sequence Diagram



Pengguna menekan discovery, kemudian sistem aplikasi akan mengarahkan ke discovery page sehingga pengguna dapat melakukan eksplorasi terkait barang yang mereka cari atau butuhkan. Pengguna dapat melakukan pencarian dengan menekan tombol search lalu sistem aplikasi akan memproses pencarian, jika barang yang dicari ditemukan maka akan langsung ditampilkan ke pengguna, jika tidak sistem aplikasi akan menampilkan saran barang yang berkaitan dengan pencarian si pengguna.

Class Diagram



D. Alloy

```
sig Login {  
  username: String,  
  password: String,  
  email: String,  
  phoneNumber: String,  
  discPage: DiscoveryPage  
}
```

```
sig DiscoveryPage {  
  itemName: String,  
  discovery: Discovery,  
  sessionStatus: String  
}
```

```
sig Discovery {  
  isProduct: some Product  
}
```

```
sig Product {  
  itemID: String,  
  price: Int  
}
```

```
sig Batu extends Product {  
  satuanM3: Int  
}
```

```
sig Besi extends Product {  
  panjang: Int  
}
```

```
sig Semen extends Product {  
  perSak: Int  
}
```

```
pred checkSession(): Login {  
  // Implementation of the checkSession method  
  l.discPage.sessionStatus = "active"  
}  
  
pred search(dp: DiscoveryPage, itemNameToSearch: String) {  
  // Implementation of the search method  
  some p: Product | p in dp.discovery.isProduct and dp.itemName = itemNameToSearch  
}  
  
pred productInfo(p: Product) {  
  // Implementation of the productInfo method  
  p in Discovery.isProduct  
}  
  
pred getPriceBatu(b: Batu, p: Product) {  
  // Implementation of the getPrice method for Batu  
  p in Discovery.isProduct =>  
    let totalPrice = mul[b.satuanM3, p.price] |  
    totalPrice > 0  
}  
  
pred getPriceBesi(b: Besi, p: Product) {  
  // Implementation of the getPrice method for Besi  
  p in Discovery.isProduct =>  
    let totalPrice = mul[b.panjang, p.price] |  
    totalPrice > 0  
}  
  
pred getPriceSemen(s: Semen, p: Product) {  
  // Implementation of the getPrice method for Semen  
  p in Discovery.isProduct =>  
    let totalPrice = mul[s.perSak, p.price] |  
    totalPrice > 0  
}  
  
pred show {  
  all l: Login, dp: DiscoveryPage |  
    checkSession() => search(dp, dp.itemName)  
}
```

```

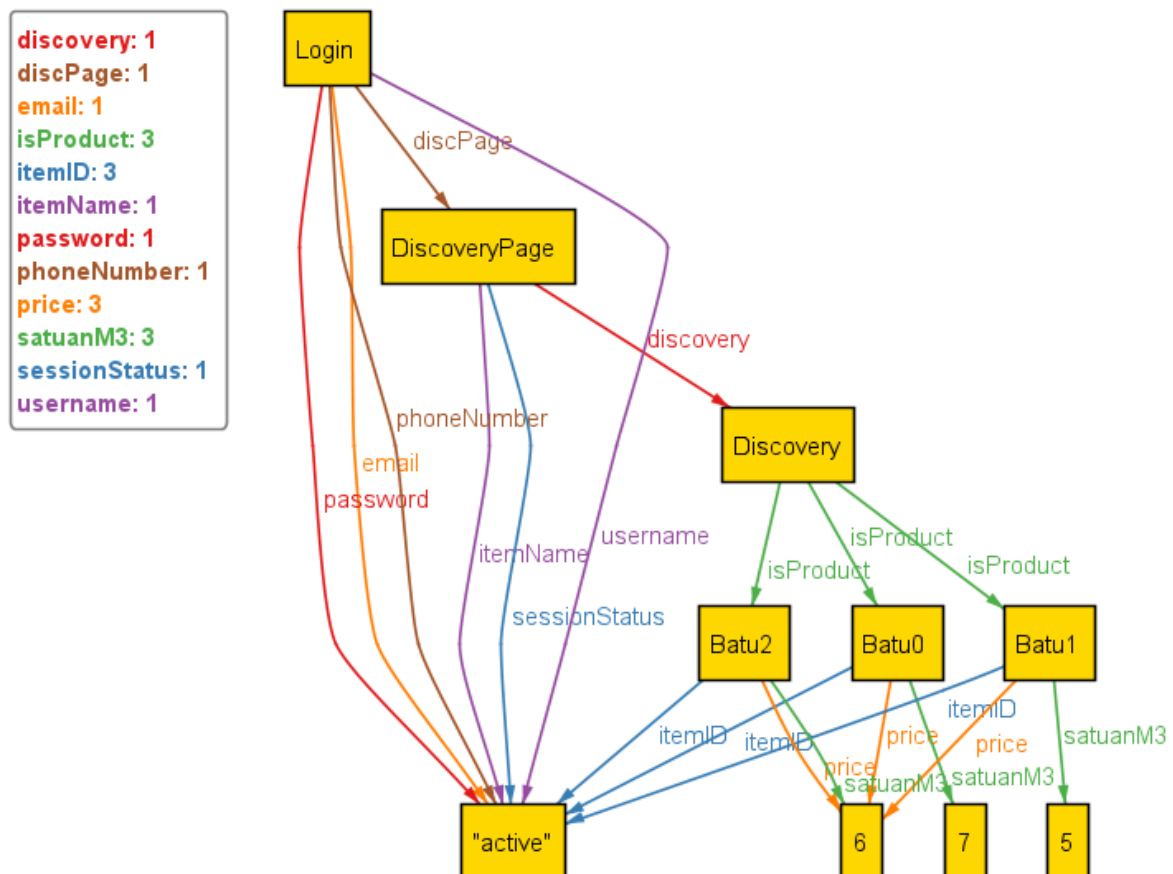
fact {
  one login: Login, dp: DiscoveryPage |
    login.discPage = dp and
    dp.sessionStatus = "active"
}

assert AssertionCheckSessionActive {
  all l: Login, dp: DiscoveryPage |
    checkSession[l] => search[dp, dp.itemName]
}

run show for 5 but 4 Product, 3 Batu, 3 Besi, 3 Semen

check AssertionCheckSessionActive

```



Executing "Check AssertionCheckSessionActive"

Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20 Mode=batch
1610 vars. 267 primary vars. 2878 clauses. 21ms.
No counterexample found. Assertion may be valid. 5ms.

Kondisi lainnya dengan dp.sessionStatus "inactive"

```
fact {  
  one login: Login, dp: DiscoveryPage |  
    login.discPage = dp and  
    dp.sessionStatus = "inactive"  
}
```

Executing "Check AssertionCheckSessionActive"

Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20 Mode=batch
1815 vars. 288 primary vars. 3238 clauses. 15ms.
Counterexample found. Assertion is invalid. 7ms.